LEARNING ARTIFICIAL GRAMMARS WITH COMPETITIVE CHUNKING

AIP-80

Emile Servan-Schreiber and John R. Anderson
Department of Psychology
Carnegie-Mellon University

August 1989

The Artificial Intelligence and Psychology Project

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

LEARNING ARTIFICIAL GRAMMARS WITH COMPETITIVE CHUNKING

AIP-80

Emile Servan-Schreiber and John R. Anderson

Department of Psychology

Carnegie-Mellon University

August 1989

We thank Don Dulany, Leigh Nystrom, Arthur Reber, Kurt VanLehn, and an anonymous reviewer for their precise comments on this paper, and Soar (Newell, in press) for some inspiration.

This paper is currently in submission for publication in the *Journal of Experimental Psychology: Learning, Memory, and Cognition.*

This research was supported in part by grant 87-51890 from the National Science Foundation, and in part by the Computer Sciences Division, Office of Naval Research, under contract N00014-86-0678. Reproduction in whole or in part is permitted for any purpose of the United States Government. Approved for public release; distribution unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; Distribution unlimited
DECLASSIFICATION / DOWNGRADING SCHEDULE	
PERFORMING ORGANIZATION REPORT NUMBER(S): IP - 80	5. MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION: Carnegie-Mellon University; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: Computer Sciences Division, Office of Naval Research
ADDRESS (City, State, and ZIP Code): Department of Psychology, Pittsburgh, Pennsylvania 15213	7b ADDRESS (City, State, and ZIP Code): 800 N. Quincy Street, Arlington, Virginia 22217-5000
NAME OF FUNDING / SPONSORING ORGANIZATION: e as Monitoring Organizatior; 8b. OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-86-K-0678
ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS p4000ub201/7-4-86

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
N/A	N/A	N/A	N/A

TITLE (Include Security Classification): learning Artificial Grammars with Competitive Chunking

PERSONAL AUTHOR(S): Emile Servan-Schreiber and John R. Anderson

TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept15 TO 91Sept14	1989 08 06	35

SUPPLEMENTARY NOTATION: mitted to the Journal of Experimental Psychology: Learning, Memory, and Cognition

COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Unintentional Learning / Artificial Grammars / Chunking / Perception

ABSTRACT (Continue on reverse if necessary and identify by block number)

SEE REVERSE SIDE

DISTRIBUTION / AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
NAME OF RESPONSIBLE INDIVIDUAL: Dr. Alan L. Meyrowitz	22b. TELEPHONE (Include Area Code): (202) 696-4302 — 22c. OFFICE SYMBOL: N00014

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

Abstract

When exposed to a regular stimulus field, for instance generated by an artificial grammar, subjects unintentionally learn to respond efficiently to the underlying structure: Miller (1958) reports that subjects memorize letter strings generated by an artificial grammar faster than randomly generated strings. Reber (1967) reports that, following rote memorization of exemplar sentences, subjects efficiently discriminate grammatical from non-grammatical strings. We explored the hypothesis that the learning process is chunking and that grammatical knowledge is implicitly encoded in a hierarchical network of chunks. Grammatical judgments are then based on the degree to which integrated representations of strings can be built using those chunks. We trained subjects on exemplar sentences while inducing them to form specific chunks. Their grammatical knowledge was then tested with a discrimination task. We found that subjects were less sensitive to grammatical violations that preserved their chunks than to violations that did not. We derived the theory of competitive chunking (CC) and found that it successfully reproduces, via computer simulations, both Miller's experimental results and our own. In CC, chunks are hierarchical structures strengthened through application during bottom-up parsing of stimuli. As stimulus redundancy creates overlapping chunks, a strength-mediated competition process determines which chunks are created and which apply during parsing.

Keywords: unintentional learning, artificial grammar, chunking, Perception (Psychology) ~~Simulation~~ Publication. (E.C.)

Learning Artificial Grammars with Competitive Chunking

The world is regular, and we are efficient regularity detectors. Some times we are intentionally looking for structural regularities. Other times, however, we learn to respond to structured stimuli even though we do not suspect an underlying structure. This latter ability, which we think is best captured by the phrase *unintentional learning*, has been most consistently studied using artificial grammars to generate regular stimulus fields. Such a grammar is shown in graphic form in Figure 1. In particular, two studies by Miller (1958) and Reber (1967) demonstrate the basic phenomenon.

Miller reports that subjects can memorize lists of letter strings generated by an artificial grammar faster than lists of randomly generated strings. While his subjects were kept intentionally ignorant of the generating principles underlying the two types of lists, they responded efficiently to the greater inter-string similarity of grammatical strings.

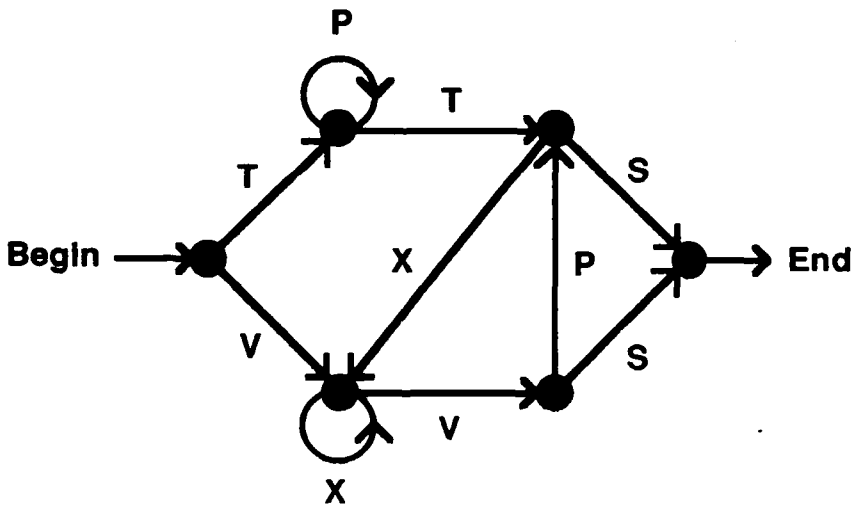

Figure 1

State diagram representation of an artificial finite state grammar. This grammar was introduced in Reber (1967), and was used to generate stimuli in our Experiment 1.

Reber elaborates on Miller's experiment by following the memorization task with a discrimination task (1967, experiment II). The combination of these two tasks is what we

henceforth refer to as the *Reber task*; The general design is as follows: Subjects are first asked to memorize some letter strings that, unbeknownst to them, are generated by an artificial grammar. After they have reached some learning criterion, the existence of the grammar is revealed, and subjects are asked to discriminate grammatical from non-grammatical strings based on their experience with the memorized grammatical strings. Reber reports that subjects are able to do so efficiently, even though their ability to verbalize their knowledge of the underlying grammar is very weak.

As Reber points out, two questions must be answered: (a) What is the form of the knowledge acquired during memorization that allows the subjects to efficiently discriminate grammatical from non-grammatical strings, and (b) how was that knowledge acquired?

Miller proposes that subjects who memorize grammatical lists "group and recode" them (p.49). Evidently this idea has its roots in an earlier article in which Miller introduces the idea of chunks (Miller, 1956). However, in both these articles Miller seems to argue that the formation of chunks is an explicit and intentional recoding process. Reber reasons that, if this were the learning mechanism, subjects' knowledge of the grammar would have to be mostly verbalizable. Since his subjects were unable to verbalize their knowledge, he rejects Miller's hypothesis of recoding and proposes the existence of an unconscious covariation detection process that yields unverbalizable, abstract grammatical knowledge.

Our own position, which we propose and defend here, is that the learning mechanism is some sort of chunking, and that the resulting knowledge on which grammatical judgments are based is a hierarchical network of chunks that, by virtue of having been created from grammatical strings, implicitly encodes grammatical constraints.

Our position differs from Miller's mainly because the concept of chunking has evolved since he introduced it thirty years ago. Whereas Miller (1956) proposes chunking mainly as a conscious recoding strategy, it is now usually understood to be a general learning mechanism not necessarily bound to the conscious and intentional realms of cognition (e.g., Newell, in press). In the absence of a good definition, we like to think of chunking as our natural, maybe automatic, tendency to process stimuli by parts.

There is a wealth of evidence that chunking is the natural learning process when the task is to memorize complex nonsense material (e.g., sequences of letters). For example, the 26 letters of the alphabet seem to be encoded into seven chunks: abcd, efg, hijk, lmnop, qrst, uvw, and xyz (Klahr, Chase, & Lovelace, 1983). Gestalt principles of proximity induce subjects to form specific chunks (e.g., Bower & Winzenz, 1969; Johnson, 1970). But even if no organization exists a priori in the stimuli, subjects impose

their own (e.g., Chase & Ericsson, 1981; Johnson, 1970; Tulving, 1962). A subject's chunking behavior is often revealed at recall by "transition-error probabilities" (e.g., Bower & Springston, 1970; Johnson, 1970), or "subjective organization" (Tulving, 1962).

Dulany, Carlston, and Dewey's (1984; 1985) analysis of the Reber task also points to the crucial role that chunking seems to play in the unintentional learning phenomenon. They propose that the basic unit of subjects' grammatical knowledge is a small group of letters; what they call a "feature". In their analysis, subjects base their grammatical judgments on a dynamically evolving collection of explicit rules of the form: *the presence of this feature implies that the string is (or is not) grammatical.* While we agree with Dulany et al. that features (we call them chunks) are the crucial units of grammatical knowledge, we disagree that such rules are needed to account for subjects' discrimination behavior.

Hence there is reason to believe that subjects faced with the task of memorizing a meaningless and long enough string of letters will chunk it into pieces. For example, if a string to be memorized is TTXVPXVS, a subject may first create the chunks (TTX), (VP), and (XVS). Then further chunking may proceed until a single chunk encodes the whole string, at which point the string is assumed to be memorized. So, for example, the two following chunks are created in succession: ((TTX) (VP)), and (((TTX) (VP)) (XVS)) which encodes the whole string. In the process of memorizing the string, the subject has created five chunks. These chunks are organized in a hierarchy at the bottom of which are *elementary* chunks which are the letters themselves. At the next level up are the *word* chunks which are made of those elementary chunks. At the top of the hierarchy are the *sentence* chunks, which encode a full stimulus. In between the word and sentence levels are any number of hierarchically organized levels of *phrase* chunks. Figure 2 represents a portion of what such a hierarchical network of chunks would look like if the above string, among others, had been chunked as specified above.

In a situation where many strings have to be memorized and they are intrinsically similar to each other (because of an underlying grammar) the chunks may reveal those similarities to some degree, thereby both facilitating and constraining further learning. For example, if the string TTXVPXVS had been memorized as specified above, and the string VXVPXXXVS had to be memorized next, some the chunks created while memorizing the former could be used to memorize the latter. Instead of perceiving the new string as just a sequence of nine letters (nine chunks), a subject may perceive it as a sequence of six chunks: V X (VP) X X (XVS). Not only does the new string immediately appear less complex, but this representation also constrains further chunking. The next chunks to be created would be (VX) and (XX) — which yields a four-chunks representation. The final representation of that string, once memorized, may then be (((VX) (VP)) ((XX) (XVS))).

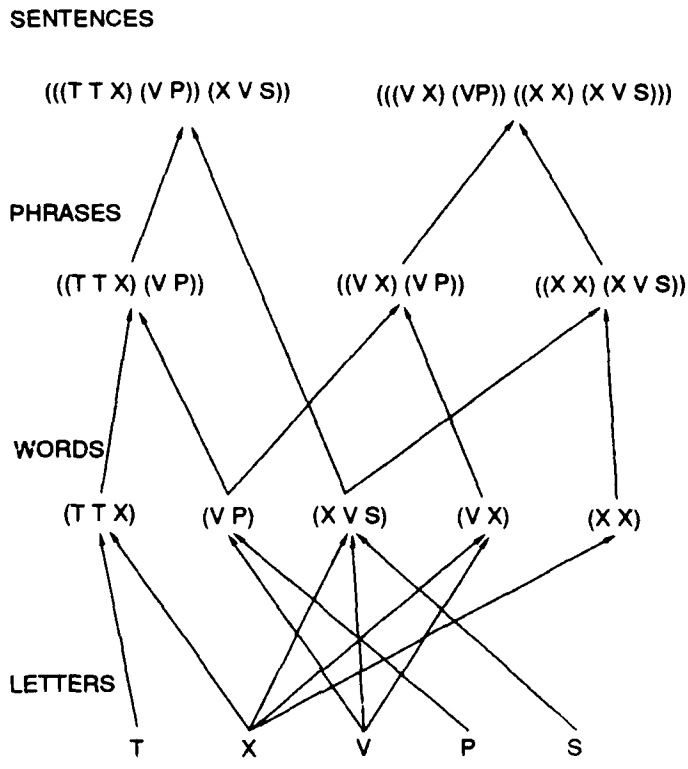

Figure 2

A hierarchical network of chunks that encodes two grammatical sentences from the grammar in Figure 1: TTXVPXVS, and VXVPXXXVS. The representations of the two sentences overlap where they share chunks (i.e., (V P) and (X V S)). The network is a multi-level structure. The bottom-most level is that of elementary chunks, in this case letters. The next level up is that of words. The next levels up are that of phrases, of which the highest level is that of full sentences.

The smaller the number of chunks that are needed to describe a string, the more familiar that string appears. The crucial variable is not the total number of chunks in the hierarchical representation of a string, but rather the number of chunks at the top level of that hierarchical representation, before any new chunks are created. So, for instance, in the situation where the string TTXVPXVS is already memorized as above, its hierarchical representation has a single chunk at the top level: (((TTX) (VP)) (XVS)). Hence that string is maximally familiar.[1] On the other hand, the new string VXVPXXXVS may be represented as V X (VP) X X (XVS) with six chunks at the top level. It is less familiar than a

[1] Indeed, the original definition of a chunk in Miller (1956) is that of a "familiar" unit of knowledge (p. 93).

memorized string, but more familiar than it would have been if no word chunk had transferred to its representation (thus yielding nine chunks at the top level, one per letter.) We call that crucial number of chunks *nchunks*. We tentatively define *familiarity* as:

$$e^{1 - \text{nchunks}} \quad (1)$$

Hence the familiarity of a stimulus ranges from 1 (maximally familiar) to an asymptotic 0 (maximally unfamiliar).[2]

In summary, our hypothesis was that during the memorization task subjects build a hierarchical network of chunks in which, due to the redundancy inherent in a grammar-generated set if strings, the representations of the strings overlap where they share chunks. The familiarity of a string increases with the degree to which an integrated representation of that string can be built from the existing chunks.

We hypothesized further that the probability that a string is judged grammatical increases with how familiar it appears given the network of chunks acquired during the memorization task. Hence, the more existing chunks are preserved in a string, the more chance it has of being judged grammatical.

In order to test this hypothesis we had to control which chunks subjects created during the memorization task. Then we could test their grammatical knowledge by including in the test basically two sorts of non-grammatical strings: strings which violated the grammar while preserving subjects' chunks, and strings which violated the grammar without preserving subjects' chunks. Our hypothesis directly predicted that strings of this this latter sort would appear less familiar, and therefore tend to be rejected more often, than strings of the earlier sort.

We tested this hypothesis with two analogues to Reber's (1967) experiment. The basic design of both our experiments was the same. The only significant departure from the original Reber task design was our addition of experimental conditions in which subjects were strongly induced to form specific chunks instead of being left to chunk the strings by themselves. The first experiment tested our hypothesis at the level of word chunks, and the second tested it at the level of phrase chunks. In each experiment the non-grammatical strings included in the test either preserved the chunks (words in Experiment 1, words and phrases in Experiment 2), or did not preserve them (in Experiment 2, those strings preserved only the words).

---

[2]This definition has not been experimentally derived. However, our intuition is that familiarity must be a rapidly decreasing function of nchunks.

Experiment 1

Method

Stimulus Materials. The artificial grammar we used to generate the stimulus set is represented in Figure 1. It is the one used by Reber (1967). There are exactly 34 grammatical strings of length six to eight in this language. We chose 20 of those to be memorized by subjects. For clarity, we will henceforth reserve the word "sentence" to refer to grammatical strings as opposed to non-grammatical strings. It is also important to note that whenever a string is referred to as grammatical or not, it is relative only to the grammar we used to generate the set of training strings (*E's grammar*, in Figure 1), not relative to any of the numerous other grammars which could also have generated that set.

Subjects and Conditions. The subjects were 37 Carnegie-Mellon University undergraduates, participating as a requirement in an introductory psychology course, and receiving five dollars.

There were four conditions with ten subjects in one and nine in each of the other three. Three of the subject groups (Well-structured-1, Well-structured-2, and Badly-structured) saw sentences in which groups of letters were separated by spaces. Johnson (1970) demonstrates that the Gestalt principle of spatial proximity is very effective in biasing subjects toward forming specific chunks. The remaining group of subjects (Unstructured) saw the same sentences but with no a priori organization into words, just sequences of letters. These subjects were in the same situation as Miller's (1958) and Reber's (1967) subjects.

The point of having three different conditions in which the sentences were structured was to investigate the effects that different structurings would have on subject performance in both the memorization and judgment tasks. In the two well-structured conditions, the chunks were designed to make the similarities among the sentences more apparent than in the unstructured condition. In contrast, in the badly-structured condition, the chunks were designed to make these similarities more difficult to notice. In all four conditions, the 20 sentences presented for memorization were the same but for the change in format.

String format in the well-structured-1 condition. Reber (1967) noted that E's grammar (see Figure 1) can be expressed as the union of five sentence types: (1) T (P)* TS, (2) T (P)* TX ((X)* | VPX)* VS, (3) T (P)* TX ((X)* | VPX)* VPS, (4) V ((X)* | VPX)* VS, (5) V ((X)* | VPX)* VPS. This way of structuring the grammar yields a set of initial, middle, and final words on which the alternative representation of E's grammar in Figure 3a is based: *initial words* are T, and V; *middle words are* TX, VPX, (P)*, and (X)*; *final words* are TS, VS, and VPS. A total of nine words.

String format in the well-structured-2 condition. Another way, among many, of expressing E's grammar in terms of a set of words is given in Figure 3b (this grammar is equivalent to E's grammar for all strings of more than four letters). It is based on a total of 11 words: Initial words are $T(P)^+T$, TT, $TT(X)^+$, VVP, V, and $V(X)^+$; middle words are VP, and $(X)^+$; final words are S, $(X)^+VS$, and VPS. This particular structuring of E's grammar has the following two properties: (a) no word is of length larger than three (assuming that the length of a run is one), and (b) the average length of a sentence, in words, is close to three when the length in letters is limited to eight (as is the case of the training sentences). Our choice of the number three is based on its mention as "the largest size chunk everyone is willing to use" (Johnson, 1970, P. 211).

String format in the badly-structured condition. In contrast with the two well structured formats, the badly structured format was very unsystematic. It was designed to have as many words as possible, within the constraint that no sentence (in the memorization set) should have more than three words. By minimizing the transfer of words among the sentences this structuring made the similarity of the sentences non-apparent. The sentences were based on 27 words (versus 9 and 11 in the well-structured-1 and well-structured-2 conditions respectively): Initial words were $T(P)^+$, $V(X)^+$, VV, V, T, TTX, TT, VVP, and TPT; middle words were $(X)^+VP$, $(X)^+V$, $P(X)^+$, $T(X)^+$, PTX, VP, XX, VPXV, XVPX, PPPP, and VPX; final words were $(X)^+VS$, PTS, S, VPS, VS, PS, and TS.

As an example of how the different structurings made the strings look more or less similar to each other, Table 1 shows how the two strings TPPTXXVS and TPPPTXVS were differently formatted in each of the four conditions.

Table 1

Examples of how the same strings were structured differently in the four conditions

	Condition		
unstructured	well-structured-1	well-structured-2	badly-structured
TPPTXXVS	T PP TX X VS	TPPT XXVS	TP PTX XVS
TPPPTXVS	T PPP TX VS	TPPPT XVS	TPPP TX VS

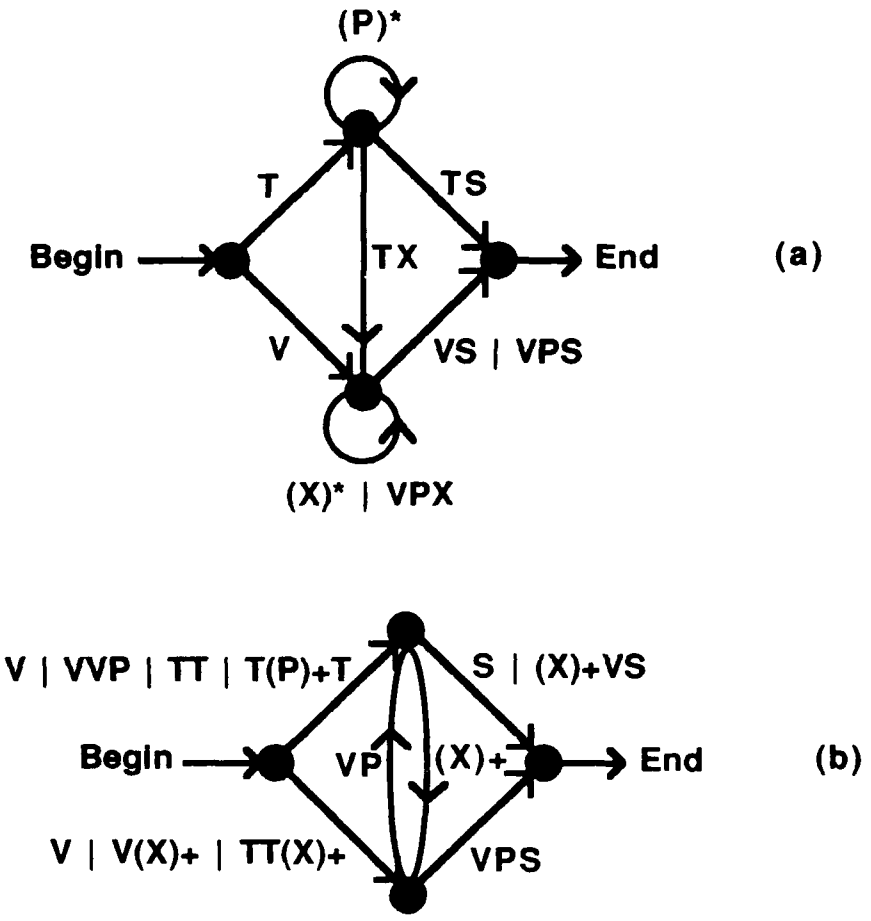

Figure 3

Two alternative, word-based, representations of E's grammar (see Figure 1): (a) was used to structure the sentences of E's grammar in the well-structured-1 condition, and (b) was used to structure the sentences in the well-structured-2 condition.

The memorization task (training). The 20 sentences chosen for the memorization task were distributed in 5 sets of 4. Subjects were asked to memorize each set, numbered 1 through 5, in that order. However, once they had memorized a set they were told that they did not have to try to remember its sentences anymore, but to just concentrate on memorizing the sentences in the next set. The sentences in a set were presented individually for 5 seconds on a computer screen, and in a fixed order. After viewing the four sentences once (one trial), subjects were asked to reproduce the set by

writing on a blank card. They had to achieve two correct reproductions in a row before moving to the next set, without delay. The sets (included in full in the appendix) were identical in the four conditions, except for the changes in sentence format. The sets were designed so that four of the five different sentence types of the well-structured-1 format were represented in each set. Subjects were not informed of the true nature of the sentences, which were referred to simply as "strings", and the task was presented as a rote memory experiment.

The discrimination task (testing). Immediately after completing the memorization task, subjects were told that the twenty sentences they had memorized were all examples of "good strings"; they would now be asked to judge whether many other strings were "good" or "bad", based on their experience with the training strings. They were not asked to give any reasons for their answers, but were given the following general clues about what could make a string "bad": (1) something may be missing, (2) something may be extra, and (3) the order of letters in the string may be wrong. As the strings presented in this task were all in unstructured format, subjects in the three structured conditions were warned of the format change. All the subjects were tested on the same strings. There were 228 strings in this task: 82, or 36%, were grammatical (G), and 146, or 64%, were non-grammatical (NG). Subjects were not told about these proportions. Each string was presented individually on a computer screen for as long as it took a subject to make a judgment. No feedback on the correctness of the judgments (with respect to E's grammar) was given.

Grammatical strings. There were two types of grammatical strings in the test. The *Old-grammatical* strings were the 20 sentences which the subjects had just memorized. The *New-grammatical* strings were 21 newly generated sentences. Each was presented twice during the test. The length of the sentences varied from four letters to eight letters.

Non-Grammatical strings. The 146 non-grammatical strings, six to ten letters long were all different (no repetitions). The words of the well-structured-1 format had a special role in the design of these strings. Specifically, the non-grammatical strings either preserved or did not preserve these words; a non-grammatical string could either be parsed using only these words, or some part of it could not be parsed using only these words. There were five types of non-grammatical strings, two of which preserved the well-structured-1 words (*all-words strings*) and three of which did not (*non-words strings*).

All-words strings were non-grammatical because they violated the word-order constraints of the grammar. The violation occurred either at the end of a string or elsewhere . To generate such a string we first generated a grammatical string, structured it in terms of well-structured-1 words, and then either deleted a word, added a word, or

replaced one word with another. If, as a result, the last word of the string was not one of the three valid final words of the well-structured-1 format, then the string was of the *all-words/bad-final* type. Else, it was of the *all-words/good-final* type.

Non-words strings were of one of three types. The *non-words/random* type strings were either completely randomly generated from the five letters of the grammar, or had a randomly generated middle part framed by valid initial and final words. In the two other types — *non-words/bad-final* and *non-words/good-final* — the strings were generated as follows: First, a grammatical string was generated and structured in terms of well-structured-1 words. Then, one of these words was made a non-word by simply replacing one of its letters with another. If the non-word was the last word of the string then the string was of type non-words/bad-final. Else it was of type non-words/good-final.

It seemed important to explicitly distinguish violations occurring at the end of a string from violations occurring elsewhere because other researchers have reported that, at least with a similar grammar, subjects are especially sensible to violations occurring at the end of a string (e.g., Reber, 1967; Reber & Lewis, 1977). There were 88 strings of type all-words/good-final, 30 strings of type all-words/bad-final, 8 strings of type non-words/good-final, 8 strings of type non-words/bad-final, and 12 strings of type non-words/random.[3]

Order of presentation of the test strings. The order of presentation of the test strings was randomized once with two constraints: (a) that each grammatical string appears once in the first half, and a second time in the second half of the presentation sequence, and (b) that an equal number of strings of each non-grammatical type appears in each half of the presentation sequence. The order of test strings was then the same for every subject. Subjects were told that the length of a string is not an indication of its "goodness", although they had only seen strings of lengths six to eight in the memorization task. They were also warned about the presence of strings they had already seen (hence "good"), and of the repetition of some strings, but not that only the "good" strings would appear twice.

Specific Predictions

The memorization task. If, as we hypothesized, subjects in the unstructured

---

[3]This striking imbalance in the numbers of strings per types results from the original exploratory nature of this experiment. When it was designed we had no really precise hypothesis about subjects' behavior in the Reber task, except that it should reflect some chunking activity. 22 different types of non-grammatical strings, generated from 22 possible operations involving initial, middle, and final words, alone or in combination, were each represented by six to ten strings in the test. Only later, when our theory became precise enough, did it also become clear that the fundamental tests of it rested in the comparisons of the all-words vs. non-words types of strings.

condition chunk the sentences themselves, then they should not be at a significant disadvantage compared to the subjects in the two well-structured conditions. Hence their ease of memorizing the twenty exemplar sentences should be comparable to that of subjects in these well-structured conditions. On the other hand, subjects in the badly-structured condition should have more difficulty since their sentences are structured so as to minimize the transfer of chunks among sentences.

The discrimination task. Our general prediction, that subjects would reject more the strings that did not preserve their chunks than the strings which did preserve them, was testable only within the well-structured-1 condition since the non-grammatical strings were especially designed to either preserve or not preserve word chunks acquired by these subjects. The specific prediction was that subjects in the well-structured-1 condition would reject the strings of type non-words/bad-final more than the strings of type all-words/bad-final, and would reject the strings of type non-words/good-final more than the strings of type all-words/good-final.

Results

The memorization task (training). We had intended, before any subject was run, to look for indirect evidence of chunking in the written protocols of subjects in the unstructured condition. However, this proved unnecessary; these subjects had a strong tendency to *overtly* chunk the training sentences, i.e., to reproduce a sentence as separate groups of letters (e.g., reproducing TTXXXVS as TT XXX VS), or to write the end of a sentence before writing its beginning. More precisely, we computed how many of the 20 training sentences were overtly chunked at least once (on a correct reproduction) by each subject in this condition. We found that, on average, these subjects overtly chunked 14.5 of the 20 training sentences, i.e., 72.5%. This does not mean that these subjects did not chunk the remaining 27.5% of the training sentences, but just that they did not do so overtly. Such direct evidence of chunking is, for our present purpose, the crucial result we extracted from the protocols.

There were systematic regularities in how many presentations of each set were necessary for subjects to reach criterion. These results are plotted in Figure 4. An ANOVA including the four conditions and the five successive sets revealed a main effect of condition, $F(3, 33) = 8.54$, $MSe = 14.23$, $p < .001$, and a main effect of set number, $F(4, 132) = 5.52$, $MSe = 3.81$, $p < .001$. Furthermore, sentence format interacted significantly with the ease of learning of individual sets, $F(12, 132) = 2.13$, $MSe = 3.81$, $p < .025$. To check for the effect of presenting the strings in well-structured versus unstructured format, we did a two conditions by five sets ANOVA in which the conditions

were unstructured and well-structured (i.e., the well-structured-1 and well-structured-2 subjects were grouped together). We found no significant main effect of condition, $F(1, 26) = 1.91$, $MSe = 15.62$, $p = .18$, or of set number, $F(4, 104) = 2.02$, $MSe = 4.02$, $p = .097$, and no interaction, $F(4, 104) = .46$, $MSe = 4.02$, $p = .76$ Hence, we concluded that the main effect of condition found when the four conditions were included in the analysis was essentially due to the poorer performance of the subjects in the badly-structured condition. As predicted, the manipulation of inducing subjects to form specific chunks in the two well-structured conditions, instead of letting them decide which chunks to form (unstructured condition), did not yield any significant advantage in the memorization task.

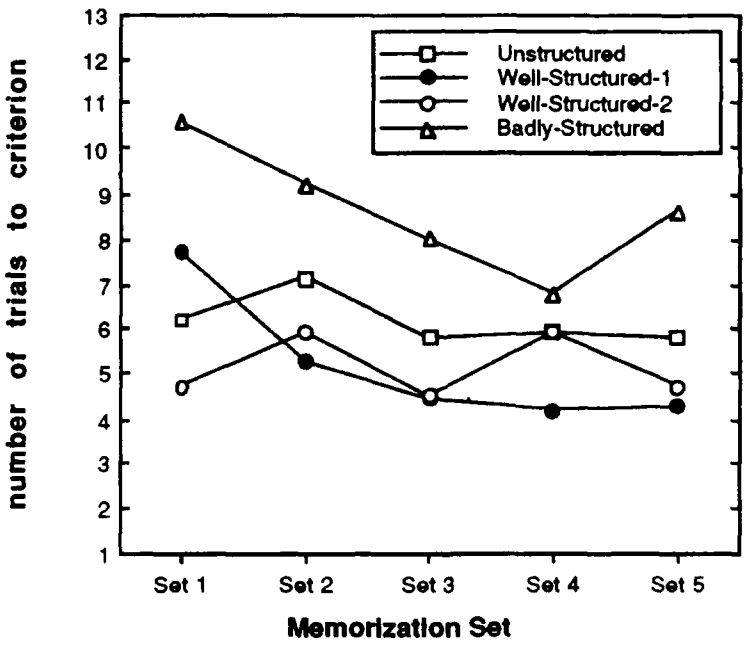

Figure 4

Mean number of presentation trials needed to reach criterion as a function of training set number (Experiment 1).

The discrimination task (testing). Table 2 contains the mean percentages of strings correctly classified by the subjects in each condition, and the same data broken down into correct acceptance of grammatical strings and correct rejection of non-grammatical strings. A four conditions by two types of strings ANOVA on these data revealed a main effect of condition, $F(3, 33) = 6.42$, $MSe = 100.11$, $p < .005$, that was caused essentially by the

poorer performance of the subjects in the badly-structured condition, and a main effect of the grammatical status of strings, $F(1, 33) = 17.05$, $MSe = 270.90$, $p < .001$, that reflected the fact that in all conditions the grammatical strings were more accepted than the non-grammatical strings were rejected. No interaction was revealed, $F(3, 33) = .59$, $MSe = 270.90$, $p = .63$.

Table 2

Mean percentages of correct classifications

	Condition			
	W-S-1	W-S-2	U	B-S
Percentage of correct classifications	68.8 5.7	74.8 9.0	68.9 7.1	59.1 7.7
% of grammatical strings accepted	78.7 5.8	83.4 10.8	75.7 19.3	74.6 16.3
% of non-grammatical strings rejected	63.5 9.5	69.9 16.4	65.2 16.6	50.3 7.4

Note: The names of the conditions are abrieviated as follows: W-S-1 = well-structured-1; W-S-2 = well-structured-2; U = unstructured; B-S = badly-structured. The standard deviations from the means are shown below the means.

More fine grained results are included in Table 3. It shows the mean percentages of strings rejected in each of the seven types of strings by the subjects in each condition. Within the well-structured-1 condition, both our predictions were verified. These subjects rejected the non-words/good-final strings significantly more than the all-words/good-final strings, $t(8) = 3.6$, $p < .01$. They also rejected the non-words/bad-final strings significantly more than the all-words/bad-final strings, $t(8) = 2.5$, $p < .05$. In none of the three other conditions did these same comparisons reveal significant differences.

Another interesting result is that the subjects in the well-structured-1 condition rejected the all-words/good-final strings more than the new-grammatical strings, $t(8) = 4.7$, $p < .005$. Since the new-grammatical strings were more likely than the all-words/good-final strings to preserve the phrase chunks of these subjects (in addition to their word chunks), that difference may indicate that the subjects were also sensitive to

the preservation of their phrase chunks.

Table 3

Mean percentages of strings rejected in each of the seven string types.

String Type	Condition W-S-1	W-S-2	U	B-S
Old-Grammatical	13.9	15.8	18.3	19.7
	6.0	12.4	15.2	17.7
New-Grammatical	29.1	17.1	30.4	30.9
	7.7	12.3	24.4	18.4
All-words/Good-final	50.1	63.9	56.9	42.2
	10.8	19.4	19.0	7.8
Non-words/Good-final	73.6	63.8	54.2	40.3
	22.9	32.0	25.0	13.7
All-words/Bad-final	78.5	76.3	78.9	65.2
	21.3	22.8	21.1	15.3
Non-words/Bad-final	97.2	82.5	84.7	59.7
	5.5	18.8	16.3	21.4
Non-words/Random	95.4	92.5	87.0	73.1
	8.4	8.3	11.1	19.5

Note: The names of the conditions are abrieviated as follows: W-S-1 = well-structured-1; W-S-2 = well-structured-2; U = unstructured; B-S = badly-structured. The standard deviations from the means are shown below the means.

Discussion

It could have been argued that our main experimental manipulation of presenting the training sentences already structured, instead of unstructured as in the Miller (1958) and Reber (1967) experiments, significantly altered the nature of the memorization task. If that were true, then we could not extend our analysis of the subjects in the well-structured conditions to the subjects in the unstructured conditions. However, our results not only confirmed that the subjects in the unstructured conditions chunked the

sentences themselves, i.e., imposed their own structure, but also that their discrimination performance matched that of the subjects in the well-structured conditions. Hence, we are justified in extending our analysis of the subjects in the well-structured-1 condition to the subjects in the well-structured-2 and unstructured conditions. We must, however, consider the possibility that the subjects in the badly-structured condition may have used more complex memorization strategies than simple chunking, for the poor structure of their training sentences made this task more difficult for them than for the other subjects. Whenever an effect of condition was found, in either the memorization or discrimination task, it was always due to the poorer performance of these subjects.

Our hypothesis was that subjects chunk the exemplar sentences and then base their judgments of grammaticality on the degree to which integrated representations of strings can be built with their chunks. We found accordingly that subjects were more prone to reject strings which did not preserve their word chunks than strings which did: the subjects in the well-structured-1 condition rejected the non-words/good-final strings more than the all-words/good-final strings, and rejected the non-words/bad-final strings more than the all-words/bad-final strings. Furthermore, there was an indication that the degree to which the phrase chunks were preserved also affected judgments of grammaticality: the subjects in the well-structured-1 condition rejected the all-words/good-final strings more than the new-grammatical strings (which are more likely to preserve phrase chunks).

However, this last difference may alternatively result from the fact that the new-grammatical strings were repeated while the all-words/good-final strings were not. To eliminate this alternative explanation we designed our second experiment to test specifically for the effect of preserving phrase chunks. The basic design was the same as that of our first experiment. Exemplar sentences were memorized in different formats, and a discrimination task followed. The difference was that instead of inducing different groups of subjects to form different word chunks, we induced them to form the same word chunks but different phrase chunks. Whereas the test strings all preserved the word chunks of all subjects, they either preserved or did not preserve their phrase chunks. We predicted that subjects would tend to reject the strings which preserved only their word chunks more than the strings which preserved both their word chunks and their phrase chunks.

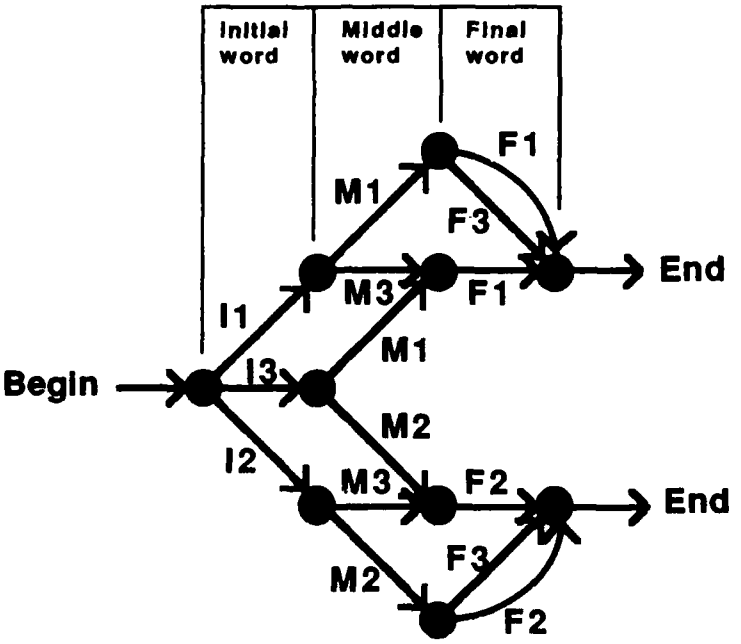

Figure 5

The grammar used to generate sentences in Experiment 2. Each of the nine symbols could be instantiated by two two-letter words. (for instance, I1 could be either FK or MV.)

Experiment 2

Method

Stimulus Materials. For this experiment we created a grammar of a different kind, one more immediately defined in terms of word sequences than in terms of letter sequences. The grammar we designed is depicted in Figure 5. Each of the nine symbols in the figure could be instantiated by two different words. The 18 possible words were meaningless pairs of consonants (e.g., FK). There were three sorts of words, as there are three sorts of symbols in the figure: initial words (I1, I2, I3), middle words (M1, M2, M3), and final words (F1, F2, F3). Every one of the 64 possible sentences was three words long. Each was made of an initial word, followed by a middle word, followed by a final word. The possible sentences could be classified into four types. Sentences of type 1 instantiated [I1 M1 F1], sentences of type 2 instantiated [I2 M2 F2], sentences of type 1/3 instantiated either [I1 M1 F3] or [I1 M3 F1] or [I3 M1 F1], and sentences of type 2/3 instantiated either [I2 M2 F3] or [I2 M3 F2] or [I3 M2 F2]. Note that sentence types 1/3 and 2/3 could be viewed as one-word distortions of the prototype sentence types 1 and 2 respectively.

Subjects and Conditions. There were two experimental conditions — *initial-middle* and *middle-final* — and a control condition. As in Experiment 1, they differed in the way sentences were presented during the memorization task. Subjects were 38 Carnegie-Mellon University undergraduates participating for credit in an introductory psychology course, and receiving three dollars. There were 14 subjects in condition initial-middle, 12 in condition middle-final, and 12 in the control condition.

The memorization task (training). Sentence presentation was manipulated in the two experimental conditions to induce specific phrase-chunkings of sentences. In condition initial-middle, we wanted subjects to chunk sentences as ((Initial Middle) Final) structures, whereas in condition middle-final, we wanted subjects to chunk sentences as (Initial (Middle Final)) structures. As in the first experiment, we used spacing between words to induce the first-order (word) chunking. The new problem was to find some mode of presentation which would get subjects to chunk the desired phrases. Our solution was to first present only the desired phrase, and then the whole sentence. For instance, to get initial-middle subjects to chunk the initial and middle words of FK TM KS together, we first presented FK TM //.[4] Only after they had memorized that did we present the whole sentence. Hence they had no choice as to which phrases to chunk. In that case, the sentence would end up being memorized as ((FK TM) KS). In contrast, middle-final subjects first had to memorize \\ TM KS before they could memorize the whole sentence. Hence their representation of the sentence would be (FK (TM KS)). Finally, control subjects were left to decide for themselves how to chunk the sentences (like the unstructured condition subjects in Experiment 1). They always saw complete sentences on the memorization trials.

The 48 sentences of types 1/3 and 2/3 were presented in the memorization task (The 16 sentences of type 1 and 2 were reserved for the discrimination task.) They were distributed in 16 sets of 3 sentences each. The distribution of sentences among sets was different in each condition, but there were always 9 different words per set, so that every subject saw all 18 words every two sets. As in Experiment 1, the sentences of a set were presented individually for 5 seconds on a computer screen. After having seen the 3 sentences of a set, subjects were asked to type them back using the computer's keyboard. The protocols were recorded directly by the experiment's program. Whereas control subjects saw complete sentences on every trial, experimental subjects only saw two words per sentence until they could correctly reproduce that. Only then did the full-sentence trials begin for a specific set. One correct reproduction of three full sentences

[4]Subjects were told that "//" was simply a place-holder for a future word.

was required before moving to the next set. As in Experiment 1, subjects were not informed of the true (grammatical) nature of the sentences, which were referred to simply as "strings", and the task was presented as a rote memory experiment.

The discrimination task (testing). Immediately after having mastered the sixteenth training set, subjects were told that they had just seen 48 "good" strings, and that they would now have to try to discriminate between new "good" strings and "bad" strings. They were told that the good strings would be those in which the three words seemed to "fit well together".

Grammatical sentences. The grammatical sentences in the test were the 16 sentences of types 1 and 2. Note that although the specific three word combinations which those sentences represent had not been seen during the training, every pairwise combination of words in such sentences had been seen twice during training (in two different sentences). Thus, the three words in a type 1 or 2 sentence would presumably seem to "fit well together". Since none of these strings had been seen during training, they were collectively referred to as *new-grammatical*. Each was repeated once.

Non-grammatical strings. These strings were designed to contain non-grammatical word pairs. There were three types of such strings: *replace-initial*, *replace-middle*, and *replace-final*. To build one of these strings, we took a type 1 or type 2 sentence (new-grammatical) and simply replaced one of the words with another of the same sort, but which had never occurred with the other two in the memorized sentences. So, for instance, to build a replace-initial string we took for example a type 1 string and replaced the initial I1 word with an initial I2 word. The resulting string instantiated [I2 M1 F1], where the initial word had never occurred with any of the other two words in the memorized sentences. Replace-middle and replace-final strings were built similarly, by replacing middle or final words. All the 48 possible replace-initial, replace-middle, and replace-final strings were included in the test (16 in each type). Thus, the subjects were to judge 80 strings (16x2 grammatical and 48 non-grammatical).

Specific predictions

Given this specific experimental design, and our general prediction that subjects will tend to reject the strings that do not preserve their chunks more than the strings that do, we predicted that initial-middle subjects would tend to reject replace-initial and replace-middle strings more than replace-final strings (which preserved their initial-middle phrase chunks). In contrast, we predicted that middle-final subjects would tend to reject replace-middle and replace-final strings more than replace-initial strings (which preserved their middle-final phrase chunks).

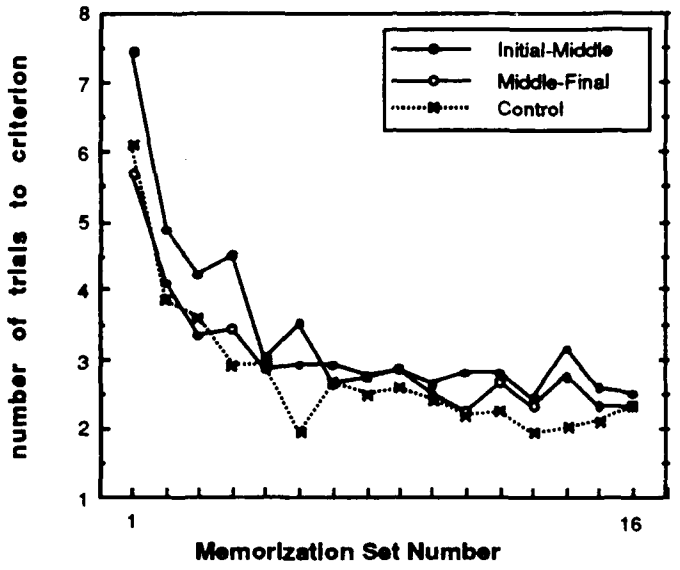

Figure 6

Mean number of presentation trials needed to reach criterion as a function of training set number (Experiment 2).

Results

The memorization task (training). The learning curves of the three subject groups are plotted in Figure 6. The only significant effect was that of trial number, $F(15, 525) = 29.75$, $MSe = 1.44$, $p < .0001$. Apparently, forcing the experimental subjects to process the sentences by parts made them neither significantly slower nor faster than the control subjects to memorize the sentences.

The discrimination task (testing). Table 4 contains the mean percentages of strings rejected in each of the four types of strings by the subjects in each condition. A three conditions by four types of strings ANOVA revealed a main effect of string type, $F(3, 105) = 12.87$, $MSe = 4.82$, $p < .0001$, due to the lower rejection level, across conditions, of the new-grammatical strings compared to the non-grammatical strings. There was also a main effect of condition, $F(2, 35) = 3.62$, $MSe = 15.97$, $p < .05$, due to the initial-middle subjects' overall higher rejection of non-grammatical strings. Finally, these factors interacted significantly, $F(6, 105) = 4.13$, $MSe = 4.82$, $p < .001$, suggesting that subjects in different conditions reacted differently to different types of non-grammatical strings, as expected.

Table 4

Mean percentages of strings rejected in each of the four string types.

String Type	Condition: Initial-Middle	Middle-Final	Control
New-grammatical	37.7	36.5	37.8
	12.6	11.9	10.0
Replace-Initial	67.9	38.6	48.5
	19.0	18.4	18.3
Replace-Middle	65.6	52.6	50.0
	22.7	19.1	16.0
Replace-Final	50.9	53.2	39.1
	21.6	19.7	11.3

Note: The standard deviations from the means are shown below the means.

Our two predictions were verified in the data. First, the initial-middle subjects rejected the replace-initial and replace-middle strings significantly more than the replace-final strings, $t(13) = 2.9$, $p < .025$, and $t(13) = 2.5$, $p < .05$, respectively. Second, the middle-final subjects rejected the replace-middle and replace-final strings significantly more than the replace-initial strings, $t(11) = 2.7$, $p < .025$, and $t(11) = 3.1$, $p < .025$, respectively.

It appeared also that the control subjects behaved more like the initial-middle subjects than like the middle-final subjects. They rejected the replace-middle strings significantly more than the replace-final strings, $t(11) = 2.7$, $p < .025$, and they had a tendency to reject the replace-initial strings more than the replace-final strings, $t(11) = 2.0$, $p = .068$.

Discussion

The results of Experiment 1 showed that subjects were more willing to classify as "good" the strings that preserved their word chunks than the strings that did not. The results of Experiment 2 demonstrated further that among the strings that preserved subjects' word chunks, those that also preserved subjects' phrase chunks were preferred.

Interestingly, the behavior of the subjects who were left to impose their own phrase structure on the sentences was very similar to the behavior of the subjects who were

induced to form (initial-word middle-word) phrases. This may be due to a natural bias, given that usual reading is left to right, toward forming such phrases.

If such a bias was real, then it may also explain why the initial-middle subjects were better discriminators than the middle-final or control subjects; the training in the initial-middle condition may have reinforced the natural tendency to form (initial-word middle-word) phrases, whereas the training in the middle-final condition may have fought against it. Hence, the initial-middle subjects may have been made additionally sensitive to non-preservation of their phrase chunks.

General Discussion

The results of our two experiments together provided strong support for our hypothesis that the main learning process in the Reber-task was some sort of chunking, and that grammatical discrimination was based on the degree to which integrated representations of strings could be built from the collection of learned chunks. This encouraged us to formulate a precise theory of the processes involved, both for learning and discrimination. We call it *Competitive Chunking* (CC). In the rest of this article we first describe the theory and then report two simulations of the experimental results of Miller (1958) and our own results from Experiment 1.

The theory of competitive chunking

In CC, chunks are hierarchical structures which components are chunks themselves. Chunks are used to parse stimuli bottom-up. The basis of the parsing process is the *match-compete-apply cycle*: all the chunks which match the current representation of a stimulus are retrieved. If two matching chunks overlap, then they become competitors . All the possible sets of non-overlapping matching chunks are formed and put in competition with each other. The competition yields a single winning set of chunks that is applied to the stimulus, changing its representation, and the cycle is complete. On the next cycle, chunks must match the new representation of the stimulus. The cycle is repeated until no chunk enters the competition. A number of cycle executions make up a *parsing episode*, the outcome of which is a hierarchical representation of the stimulus in terms of chunks. The crucial variable is the number of chunks at the top level of that hierarchical representation: *nchunks*. The value of nchunks is a measure of how integrated the representation of a stimulus is, and that translates into how familiar it is perceived to be (recall equation 1).

An example helps make clear the parsing process. Suppose that the set of existing chunks is:

$$\{(F\,B),\ (F\,B\,I),\ (C\,I\,A),\ (I\,C),\ (I\,A),\ ((F\,B\,I)\,(C\,I\,A))\}$$

Now suppose that the following stimulus is presented:

F B I C I A

Then the following four sets of chunks would compete against each other:

{ (F B), (C I A) }, { (F B), (I C), (I A) }, { (F B I), (C I A) }, and { (F B I), (I A) }

If the second set wins the competition, then the representation of the stimulus becomes:

(F B) (I C) (I A)

None of the existing chunks match this new representation, hence the parsing episode is complete. In that case, nchunks = 3. However, if the third set had won the competition, then the representation of the stimulus would have become:

(F B I) (C I A)

Then a single set of one chunk would have entered the competition:

{ ((F B I) (C I A)) }

The stimulus representation then would have become:

((F B I) (C I A))

In that case nchunks = 1, which is its minimum value.

The outcome of a parsing episode is used to guide the creation of new chunks. If a parsing episode yields a value of nchunks that is more than 1, then possible new chunks are proposed which build on the current representation of the stimulus. For example, in the case where the outcome was (F B) (I C) (I A), two possible new chunks would be proposed for creation: ((F B) (I C)) and ((I C) (I A)). These two candidates would compete and the winner would become a new chunk.

When the stimuli are unstructured strings of letters, candidate new chunks are generated with the following constraints: (a) word chunks can contain at most three letters, except if they encode runs, (b) phrase chunks can contain at most two words or phrases, and (c) the sub-chunks of a chunk must be adjacent. When the stimuli are structured strings (as in the three structured conditions of Experiment 1), the first constraint is relaxed to accept the given words.

Every chunk has a *strength* which reflects how often and recently it has applied in the past. The strength of a chunk increases by one unit every time the chunk is created, applied, or re-created. Strength decays with time. The decay function is the same one that ACT* (Anderson, 1983) proposes for declarative memory traces. At any point in time, the strength S of a chunk is the sum of the successive individual decayed strengthenings of that chunk:

$$S = \sum_i T_i^{-d} \qquad (2)$$

where T_i is the time elapsed since the ith strengthening, and d is the *decay parameter* (0

< d < 1).

Strength mediates the competition process both for parsing and learning: the probability that a matching chunk, or a candidate new chunk, enters a competition is a function of the average strength of its immediate sub-chunks, i.e., its *support*. This probability is given by:

$$\frac{1 - e^{-c \cdot support}}{1 + e^{-c \cdot support}} \qquad (3)$$

where *c* is the *competition parameter* ($c \in]0, +\infty[$), which determines the steepness of the probability curve.[5] Figure 7 plots this function for different values of c. Furthermore, the chunks that win competitions are those with largest support when the competition is for creation, and those with largest strength when the competition is for parsing.

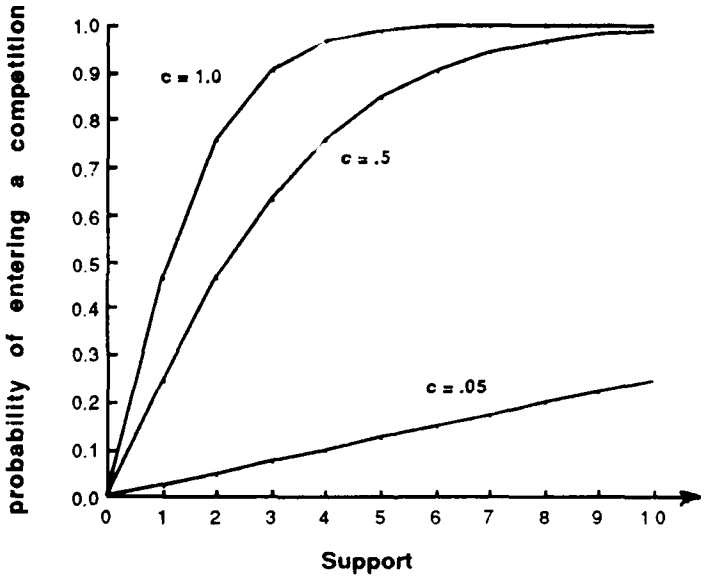

Figure 7

A matching chunk's probability of entering a competition as a function of its support (defined as the average strength of its immediate sub-chunks).

[5]The reader may recognize that this function is a sigmoid that is inflected at 0 and which values are bound between an asymptotic -1 and an asymptotic +1. However, we are considering only its positive part, since sub-strength is always positive.

An exception: since elementary chunks do not have any sub-chunks, both their strengths and supports are assumed to be constant and very large (in our simulations we set both at 10).

In each of the two simulations we report below, we made additional assumptions in order to extract from the single outcome of the parsing process — the value of nchunks — data in the same form as those collected from human subjects. For the simulation of Miller (1958) we had to transform the value of nchunks into an act of recall. In the simulation of our Experiment 1 we had to transform the value of nchunks into an act of rejection. A common simplifying assumption to both simulations was that made about time. We increased time by one unit after each parsing episode (i.e., the presentation of one string). Also, the simulations were allowed to create only one new chunk per parsing episode.

Miller-CC: a simulation of Miller's (1958) experimental results

Miller (1958) examines the effect of a stimulus field's redundancy, or lack thereof, on recall. Subjects were asked to memorize and free-recall a list of strings of letters. There were two kinds of lists. Language lists (L) were made of strings generated by the same simple finite state grammar. Random lists (R) were made of randomly generated strings (using the same letters as the grammar). A list was presented one string at a time (a few seconds each), and afterwards a subject was asked to free-recall all the strings he/she could. The dependent variable was the number of strings correctly recalled after each of ten successive presentations of a list.

There were two L-lists, L1 and L2, among which 18 strings generated from the grammar were evenly distributed (nine strings per list). There were also two R-lists, R1 and R2, with nine randomly generated strings in each. Subjects went through ten trials on one one of these lists.[6] Subjects in condition L studied L1 or L2, while subjects in condition R studied R1 or R2. Table 5 contains the the four lists used by Miller, and Figure 8 plots the recall performance of Miller's subjects. Evidently, a redundant stimulus field (L1 or L2) facilitated recall compared to a non-redundant stimulus field (R1 or R2).

In the simulation it was necessary to transform the outcome of the parsing process — the value of nchunks — into an act of recall. The assumption we made was that a string is recalled if and only if nchunks = 1 at the end of the parsing episode. The lists we used for the simulation are those in Table 5, and 40 simulated subjects were run in each condition

[6]The actual experiment had subjects study a second list after the first. Here we are mostly interested in the data obtained with the first list, so no more mention will be made of the second list.

(20 simulated subjects per list).

Table 5

Lists of strings used in Miller (1958), and the Miller-CC simulation

Redundant strings		Random strings	
L1	L2	R1	R2
SSXG	NNSG	GNSX	NXGS
NNXSG	NNSXG	NSGXN	GNXSG
SXSXG	SXXSG	XGSSN	SXNGG
SSXNSG	NNXSXG	SXNNGN	GGSNXG
SXXXSG	NNXXSG	XGSXXS	NSGNGX
NNSXNSG	NNXXSXG	GSXXGNS	NGSXXNS
SXSXNSG	NNXXXSG	NSXXGSG	NGXXGGN
SXXXSXG	SSXNSXG	SGXGGNN	SXGXGNS
SXXXXSG	SSXNXSG	XXGNSGG	XGSNGXG

Note: Reproduced with permission from Miller (1958).

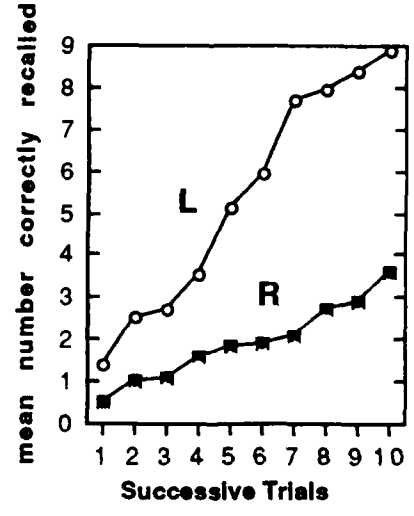

Figure 8

Mean number of strings correctly recalled by Miller's subjects on successive trials. Reproduced with permission from Miller (1958).

Many different values were tried for the two parameters c and d. Miller-CC always had an easier time learning the redundant L-lists than the non-redundant R-lists. Even when the decay parameter was set extremely close to 0 (no decay), and the competition parameter was set to a large value so that every matching chunk was sure to compete, the simple fact that less words were needed to encode the L-strings than to encode the R-strings, because of the redundancy in the L-lists, let the simulation learn the L-lists faster than the R-lists (see Figure 9a). However, a more interesting aspect of the data was more difficult to reproduce: the increase in the apparent advantage of the L-lists on successive trials. We found that this aspect of the data was reproduced qualitatively well when c = d = .5 (see Figure 9b).

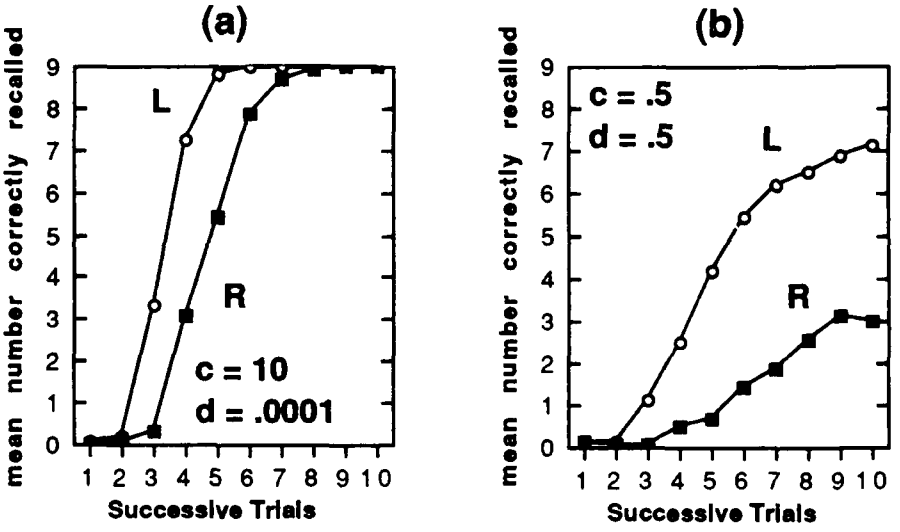

Figure 9

Mean number of strings correctly "recalled" by simulated subjects on successive trials: (a) when c = 10 and d ≈ 0, (b) when c = .5 and d = .5.

The redundancy in the L-lists is at the word level. Small groups of letters in the L-strings are shared by many L-strings. For instance, the 18 redundant strings in the L-lists of Table 5 all end with one of three triplets: XSG, SXG, or NSG. In contrast, only one final triplet is shared by two of the 18 non-redundant strings in the R-lists. Because they were in general more redundant, the words chunked in the L-strings accumulated more strength more rapidly than the words chunked in the R-strings. This, in turn, made the phrases chunked in the L-strings easier to learn and more reliably applicable than the phrases chunked in the R-strings. Hence the former were able to start to accumulate strength sooner, and to accumulate it more reliably than the latter. In the same manner,

stronger phrases chunked in the L-strings made it possible to learn entire L-strings sooner and to apply them more reliably. The strength construct and the way that it mediates parsing and learning made it possible to transfer the word-level advantage of redundancy in the L-lists to the phrase level and finally to the sentence level, eventhough the phrases and entire strings in the L-lists are no more redundant than the phrases and entire strings in the R-lists.

Reber-CC: a simulation of the results of our Experiment 1

There were 15 simulated subjects in each of our four conditions. They created chunks during the memorization task. Whereas our human subjects had to attain a criterion of two consecutive reproductions of a set before moving to the next, the simulated subjects had to "recall" (i.e., nchunks =1) the four strings of a set on the same trial once. During the discrimination task the formation of new chunks was turned off, but the strengthening and decay processes continued to operate.

To get the simulated subjects to produce discrimination data in the same form as the human subjects, we had to transform the value of nchunks into a rejection act. The assumption we made was that the probability that a string would be rejected increased with the value of nchunks. The lower the value of nchunks was, the more familiar a string appeared and the lower the probability that it would be rejected was. The simulated subjects actually accepted or rejected test strings with these probabilities. The probability function was the simple sigmoid:

$$\frac{1}{1 + e^{n - nchunks}} \qquad (4)$$

where the inflection point *n* was computed, on each individual discrimination trial, as the average value of nchunks on the previous twenty trials. Hence the same value of nchunks yielded different probabilities of rejecting a test string depending on the average value of nchunks for the few previous test strings. This means that the simulated subjects based their judgments not on how absolutely familiar a string appeared to them, but on how familiar it appeared to them relative to the average familiarity of the previous few strings. Figure 10 plots the rejection-probability function for different values of n.

Another assumption we made was that any string of letters implicitly includes extremity markers that signal the beginning and ending of that string. In the simulation we made these markers explicit in the representation of a string. Hence the representation of a string like T T X V P S was actually "begin T T X V P S end". These extremity markers were treated exactly like any single letter as elementary chunks. This assumption was important because it helped explain why subjects are apparently more sensible to grammatical violations occurring at the extremities of a string than to those occurring in the

middle of a string (Reber, 1967).[7] Those strings that have extreme groups of letters unmatched by extreme chunks can not integrate the extremity symbols in their chunked representations and hence yield increased nchunks, which increases the probability that they are rejected.

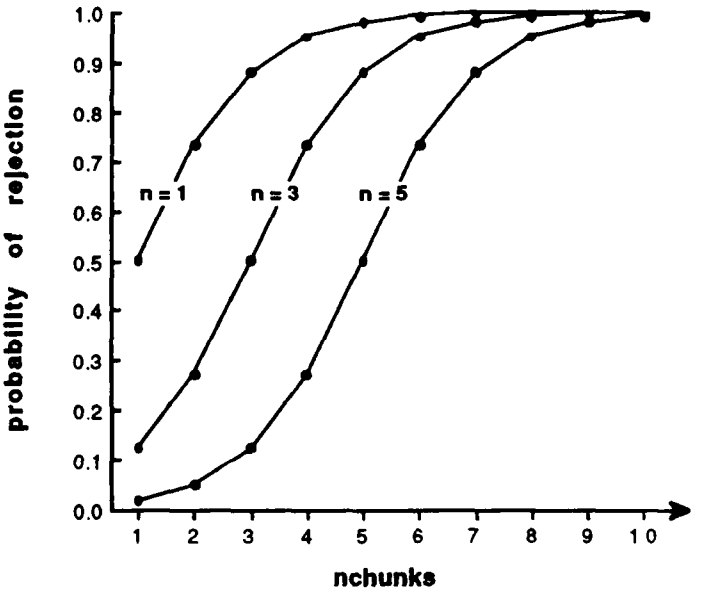

Figure 10

Probability that a string will be rejected as a function of nchunks. The sigmoid is inflected at point [n, .5] where n is computed on each trial as the average value of nchunks in the previous few trials.

The values of c and d that we used for the Reber-CC simulation were the same ones that yielded a good qualitative match to the data in the Miller-CC simulation: c = d = .5. The simulated data are plotted with the human data in Figure 11. Although the simulation was more or less successful in certain conditions and on certain types of strings, across all conditions and all types of strings the best linear fit of the simulated data to the human

[7] The set of test strings we used in experiment 1 did not allow us to test systematically for subjects' sensibility to violations occurring at the beginning of a string, although it allowed us to test systematically for enhanced sensibility to violations occurring at the end of a string. Given the grammar used in that experiment, and the format of choice for structuring and generating the non-grammatical strings (see Figure 3a), the two possible initial words were unfortunately reduced to a single letter each. This contrasted with the richness of the final words.

data had an intercept of 4.0, a slope of .95, and an r-squared of .874.

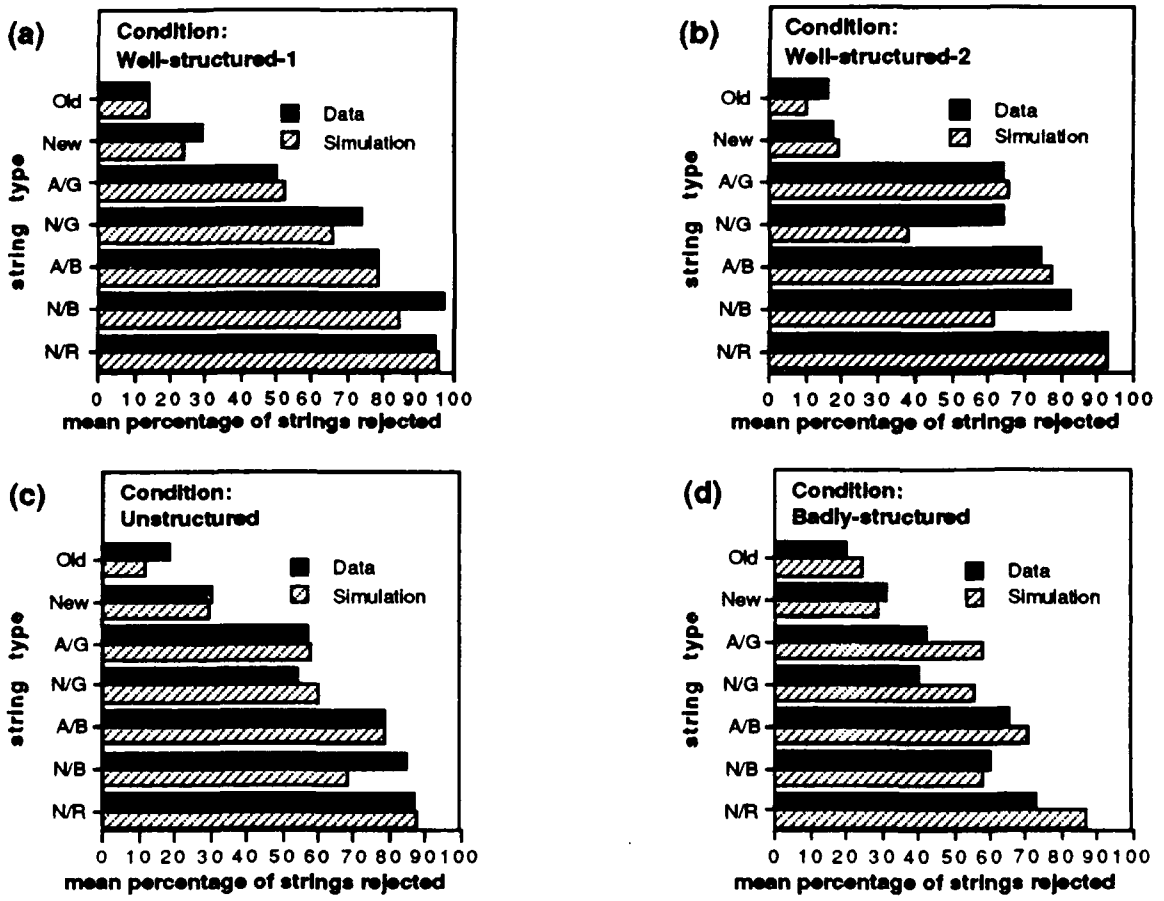

Figure 11

Human and simulated data in the discrimination task of Experiment 1: (a) in the well-structured-1 condition, (b) in the well-structured-2 condition, (c) in the unstructured condition, and (d) in the badly-structured condition. The names of the string types are abreviated as follows: Old = old-grammatical; New = new-grammatical; A/G = all-words/good-final; N/G = non-words/good-final; A/B = all-words/bad-final; N/B = non-words/bad-final; N/R = non-words/random.

Conclusion

The results of our two experiments demonstrated the important role that some sort of chunking process played in the Reber task. We built a precise theory of such a process called Competitive Chunking (CC). We found that, given a particular pair of values for its competition and decay parameters, CC was able to reproduce the important aspects of Miller's (1958) experiment on the effect of redundancy on free-recall: namely that the advantage of a redundant stimulus field increases with successive free-recall trials. Using

that same pair of values for its two parameters, and an additional process assumption for making grammatical judgments based on the output of its parsing process, CC was also able to capture 87 percent of the variance in the discrimination data of our Experiment 1 — a Reber task analogue.

Our theory is the most precise and complete theory of unintentional learning in the Reber task to date. It is a theory of (a) the learning process, (b) the representation of grammatical (i.e., structured) information, and (c) the grammatical discrimination process. The key to our success was our choice to focus primarily on the learning process. Once we had a precise theory of the learning process the theories of grammatical representation and grammatical discrimination naturally followed. That is because a learning process puts absolute constraints on the representation of what is learned, and that in turn puts important constraints on how the representation can be used. The reverse is not true however. Different types of representations, yielded by different types of learning processes, could yield the same discrimination data. Hence trying, as previous investigators have (Reber, 1967; Reber and Lewis, 1977; Dulany et al.; 1984), to derive the representation and learning processes from the discrimination data — the opposite of the route we took — could only yield mixed results.

These other investigators, however, have come to one important conclusion that we agree with: that the basic unit of grammatical knowledge in the Reber task is a small group of letters. Reber and Lewis (1977) called it a bigram or trigram covariation pattern, Dulany et al. (1984) called it a feature, and, following Miller (1958), we called it a (word) chunk. We went further and showed how these basic units of grammatical knowledge could be combined to form higher level (phrase) chunks that encode even more grammatical constraints.

In our view a subject's representation of grammatical constraints is in no way explicit. All that a subject acquires, by chunking the exemplar sentences, is a hierarchical network of chunks in which the representations of the exemplars overlap where they share chunks (see Figure 2). Grammatical constraints are implicit in this representation since the chunks were formed from grammatical sentences, and a competitive bottom-up parsing process must use these chunks to yield hierarchical representations of strings which are more or less integrated. The more integrated a representation is, the more familiar a string appears, but also the less chance there is that it violates important grammatical constraints.

It is interesting to compare our account with Dulany et al.'s. They tried to assess subjects' knowledge by asking them to explicitly mark the part of a test string that made them grammatical or non-grammatical. Subjects were supposed to underline a piece of a

string if they accepted it, or cross out a piece of a string if they rejected it. Dulany et al. considered each mark as a manifestation of a conscious rule of the form: *The presence of this group of letters implies that this string is (or is not) grammatical.* They found that the rules thus reported predicted subjects' discrimination behavior, although each was of limited scope and imperfect validity. They take this as evidence that such rules (a) exist, and (b) are not forced justifications of judgments based on more abstract representations. Our theory, of course, denies both points. Reber-CC was able to simulate the discrimination behavior of our subjects without recourse to any explict rules of the form proposed by Dulany et al. Our theory is at an advantage, however, because it accounts not only for the discrimination behavior but also for the learning by chunking behavior in the memorization task. In contrast, Dulany et al. offer no precise account of either the learning process, or the representation of the memorized exemplars. Thus the question becomes: on what basis would Reber-CC mark strings that it accepts or rejects such that these marks could be interpretted by Dulany et al. as explicit rules that predict subjects' discrimination behavior? We note that Reber-CC was never guaranteed to always reject or always accept the same string. The probability that a string was accepted really depended on how familiar it appeared relative to how familiar the previous few strings appeared. If it decided to accept a string it may decide to justify that by underlining the strongest chunk. If it decided to reject that same string, it may decide to justify it by crossing out a few adjacent letters in two adjacent chunks that are not integrated in a larger chunk.

There are striking similarities between our experiments and the language learning experiments of Morgan, Meier, & Newport (1987). Like us, those researchers studied the learning of a small artificial grammar, and induced some subjects to structure the stimuli in specific ways. However there is an important difference between their studies. Whereas their subjects were instructed to look for regularities in order to learn the underlying language, ours were kept ignorant of the existence of an underlying language and were asked merely to memorize exemplar sentences. Reber, Kassin, Lewis, & Cantor (1980) have shown that explicit instructions to look for structure significantly alter the outcome of the learning process. In our view, it must necessarily alter the learning process itself. Hence, despite their similarities, our study and Morgan et al.'s are probably not very relevant to each other. We want to make clear that although our study focussed on the learning of a simple grammatical system the theory we derived from it makes no strong claims about natural language learning. The domain of relevance of CC is more likely to be that of perceptual categorization, as typically studied using semantically poor stimuli (e.g., Posner & Keele, 1968).

References

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard university press.

Bower, G. H., & Springston, F. (1970). Pauses as recoding points in letter series. *Journal of Experimental Psychology, 83*, 421-430.

Bower, G. H., & Winzenz, D. (1969). Group structure, coding and memory for digit series. *Journal of Experimental Psychology, Monograph Supplement, 80*, 1-17.

Chase, W. G., & Ericsson, K. A. (1981). Skilled memory. In J. R. Anderson (Ed.) *Cognitive skills and their acquisition.* New Jersey: Hillsdale.

Dulany, D. E., Carlston, R. A., & Dewey, G. I. (1984). A case of syntactical learning and judgment: How conscious and how abstract? *Journal of Experimental Psychology: General, 113*, 541-555.

Dulany, D. E., Carlston, R. A., & Dewey, G. I. (1985). On consciousness in syntactical learning and judgment: a reply to Reber, Allen, & Regan. *Journal of Experimental Psychology: General, 114*, 25-32.

Johnson, N. F. (1970). Chunking and organization in the process of recall. In G. H. Bower (Ed.) *The psychology of learning and motivation, IV.* New York: Academic Press.

Klahr, D., Chase, W. G., & Lovelace, E. A. (1983). Structure and process in alphabetic retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9*, 462-477.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review, 63*, 81-97.

Miller, G. A. (1958). Free recall of redundant strings of letters. *Journal of Experimental Psychology, 56*, 485-491.

Morgan, J. L., Meier, R. P., & Newport, E. L. (1987). Structural packaging in the input to language learning: contributions of prosodic and morphological marking of phrases to the acquisition of language. *Cognitive Psychology, 19*, 498-550.

Newell, A. (in press). *Unified theories of cognition.* Cambridge, MA: Harvard university press.

Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology, 77*, 353-363.

Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior, 5*, 855-863.

Reber, A. S., Kassin, S. M., Lewis, S., & Cantor, G. (1980). On the relationship between implicit and explicit modes in the learning of a complex rule structure. *Journal of*

Experimental Psychology: Human Learning and Memory, 6, 492-502.

Reber, A. S., & Lewis, S. (1977). Toward a theory of implicit learning: the analysis of the form and structure of a body of tacit knowledge. *Cognition, 5*, 333-361.

Tulving, E. (1962). Subjective organization in the free recall of "unrelated" words. *Psychological Review, 69*, 344-354.

Appendix

The five successive training sets used in Experiment 1, in each of the four conditions

	Condition		
unstructured	well-structured-1	well-structured-2	badly-structured
Set 1			
TPPPTS	T PPP TS	TPPPT S	TPP PTS
VXXXXVPS	V XXXX VPS	VXXXX VPS	VXX XXVP S
TPPTXVPS	T PP TX VPS	TPPT X VPS	TP PTX VPS
VVPXVS	V VPX VS	VVP XVS	VV PX VS
Set 2			
VXXVPS	V XX VPS	VXX VPS	V XXV PS
TTXXXVS	T TX XX VS	TT XXXVS	T TXX XVS
TTXVPS	T TX VPS	TTX VPS	TTX VP S
VVPXXVS	V VPX X VS	VVP XXVS	VVP XX VS
Set 3			
TPPTXXVS	T PP TX X VS	TPPT XXVS	TP PTX XVS
VXVPXVPS	V X VPX VPS	VX VP X VPS	VX VPXV PS
TPPPPTS	T PPPP TS	TPPPPT S	T PPPP TS
VXXVPXVS	V XX VPX VS	VXX VP XVS	VXX VPX VS
Set 4			
TTXVPXVS	T TX VPX VS	TTX VP XVS	TT XVPX VS
VVPXXVPS	V VPX X VPS	VVP XX VPS	VV PXX VPS
VXXXXVS	V XXXX VS	V XXXXVS	VXXX XVS
TTXXVPS	T TX X VPS	TTXX VPS	TTX VS PS
Set 5			
TPTXXVPS	T P TX X VPS	TPT XX VPS	TPT XX VPS
VVPXXXVS	V VPX XX VS	VVP XXXVS	V VPX XXVS
TPPPTXVS	T PPP TX VS	TPPPT XVS	TPPP TX VS
VVPXVPS	V VPX VPS	VVP X VPS	VVP XVP S